SQL CHEATSHEET

Filter Columns

SELECT col_1, **DISTINCT**(col_2)

Unique Values

Identify Table

FROM table_1

Filter Row

Pattern Match

WHERE col_3 = 1 **AND** col_4 **LIKE** “B”

GROUP BY col_1

Aggregate Data

HAVING COUNT(*) = 2

Filter Aggregated Data

ORDER BY col_2

Order Output

COMMON OPERATORS

SELECT *c_1* **FROM** *t_1*
UNION ALL
SELECT *c_1* **FROM** *t_2*
returns the combined rows from both queries

SELECT *c_1* **FROM** *t_1*
INTERSECT
SELECT *c_1* **FROM** *t_2*
returns the intersection of both queries

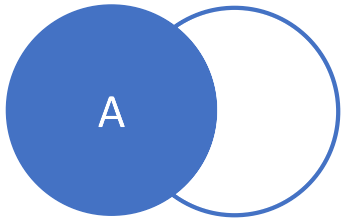
SELECT *c_1, c_2* **FROM** *t_1*
WHERE *c_1* **IS NOT NULL**
returns only rows where c_1 is not null

SELECT *c_1, c_2* **FROM** *t_1*
WHERE *c_1* **BETWEEN** 2 **AND** 20
- returns only rows where c_1 is between 2 and 20

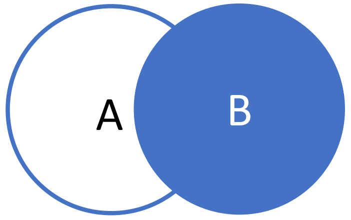
COMMON AGGREGATIONS

COUNT	return number of rows
SUM	return the sum of values
AVG	return average of the grouping
MIN	return smallest value
MAX	return largest value

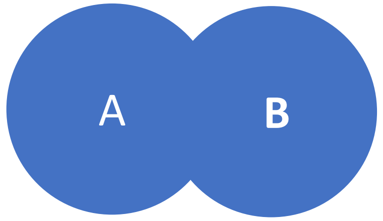
JOINS



LEFT JOIN – retain all rows from A



RIGHT JOIN – retain all rows from B



OUTER JOIN – retain all rows from A &B

WINDOW FUNCTIONS

Syntax:

SUM() **OVER** (**PARTITION BY** *col_1* **ORDER BY** *col_2*)

- returns the cumulative sum for each col_1 grouping based on col_2 ordering

Other Common Functions:

COUNT and **AVG**

ROW_NUMBER – assigns a sequential number to each row within the partitioned group

RANK – assigns a sequential number to each row skipping duplicates based on partition group

DENSE_RANK – assigns a sequential number based on partitioned group without skipping duplicates

LAG – retrieves a value based on a specified number of rows earlier

LEAD – retrieves a value based on a specified number of rows forward

NTILE – determines the percentile of a specified row within a partition

IF/ELSE

CASE
WHEN *c_1 = 1*, **THEN** *1*
WHEN *c_1 = 2* **THEN** *2*
ELSE 3 **END AS** out
FROM *t_1*

Sets the value of out based on whether c_1 is 1, 2, or something else

SELECT IFNULL(c1, 0) as c1
FROM *t_1*

Sets the value of c_1 to 0 if it is NULL

edgeGIANT

GET MORE: https://edgegiant.com/api/tools/analysis